

Examen d'informatique 2002-2003

DEUG 1^{ère} année (MASS - MIAS - SM) - Première session - Janvier 2003

Polycopiés et notes de cours autorisés - Calculatrice non autorisée

Durée : 3 heures

ARCHITECTURE DES ORDINATEURS (coefficient 2)

- a) Le bus d'un ordinateur utilise 2 octets pour transmettre les adresses. Combien d'adresses différentes peuvent être transmises ?
- b) On souhaite numériser une image en couleur avec une résolution de 500×400 pixels. Donnez une valeur approchée en Ko de la taille du fichier contenant cette image si l'on suppose que chaque pixel est codé sur 3 octets (un octet pour chacune des trois couleurs primaires) ? L'espace disponible en mémoire pour stocker cette image est limité à 50 Ko. Il faut donc procéder à la compression du fichier. Quel est le taux de compression requis ?

NUMERATION (coefficient 4)

- a) Convertir $(204, 2)_{10}$ dans les bases binaire et hexadécimale. Si le résultat a un développement infini périodique, donnez sa période.
- b) Effectuez, en base 5, la multiplication $(310)_5 \times (134)_5$ et la soustraction $(1300)_5 - (244)_5$.
- c) Soit le nombre décimal -28 . Donnez son code sur 8 bits en valeur absolue + signe, puis en complément restreint (complément à 1) et enfin, en complément à la base (complément à 2).
- d) On rappelle que les nombres entiers codés en compléments à la base sur k bits appartiennent à l'intervalle $[-2^{k-1}, 2^{k-1} - 1]$. Sur combien d'octets sont codés les nombres entiers en compléments à la base si le plus grand entier positif représentable sur une machine est égal à 32767 ?

Donnée : $32768 = 2^{15}$

COMPREHENSION (coefficient 2)

Il vous est demandé d'exécuter "à la main" le programme ci-dessous en supposant que la valeur affectée à m est égale à 9. Quelles sont les valeurs de $p1$, $p2$, $p3$ et $p4$ affichées par ce programme ?

```
program comprehension;
var i, m , p1 , p2 , p3 , p4 : integer;
begin
    readln(m) ;
    p1 := sqr(m) ;
    repeat
        p1 := p1 div 2
    until odd(p1) ;
    writeln('p1 = ' , p1) ;
```

```

p2 := m ;
while p2 > 1 do p2 := p2 mod 2 ;
writeln('p2 = ' , p2) ;
p3 := 0 ;
p4 := 0 ;
for i := 1 to m do
    if odd(i) then p3 := p3 + i
    else p4 := p4 + i ;
writeln('p3 = ' , p3) ;
writeln('p4 = ' , p4)
end.

```

PROGRAMMER UNE CALCULATRICE (coefficient 3)

Ecrire un programme en PASCAL faisant office de calculatrice, c'est-à-dire capable d'effectuer les opérations élémentaires telles que l'addition, la soustraction, la multiplication et la division. L'utilisateur du programme saisit, par l'intermédiaire du clavier, l'opération à effectuer, par exemple :

2.3+1.2

Le programme affichera le résultat :
= 3.500000000e+00

Principe de programmation :

- le programme lit le premier nombre (ici 2.3), l'affecte à une variable réelle a ;
- puis, il lit l'opération à effectuer (ici +), l'affecte à la variable caractère *oper* ;
- enfin, il lit le deuxième nombre (ici 1.2), l'affecte à une variable réelle b ;
- il effectue l'opération et affiche le résultat.

Conseil : utilisez l'instruction CASE OF.

PROGRAMMATION (coefficient 4)

Un palindrome est un texte dont la succession des lettres est la même quand on la parcourt de gauche à droite ou de droite à gauche. Un palindrome peut se composer d'un seul mot : *ressasser* ou de plusieurs mots : *engage le jeu que je le gagne*.

Ecrire la fonction **Palindrome** qui a comme paramètre un texte (chaîne de caractères) et qui retourne la valeur booléenne *vrai (true)* si ce texte est un palindrome ou *faux (false)* sinon. On suppose, pour simplifier, que le texte saisi ne contient ni caractères de ponctuation, ni symboles spéciaux. Les espaces éventuels seront supprimés avant l'appel de la fonction **Palindrome**.

Ecrire le programme principal qui effectue les tâches suivantes :

- saisie du texte et affectation à une variable x de type chaîne de caractères ;
- suppression des éventuels caractères espace : il vous est conseillé pour cela de recopier les caractères successifs de x différents de ' ' (caractère espace) dans une autre chaîne de caractères y . Ainsi, lorsque x comporte des espaces, par exemple : *engage le jeu que je le gagne*, la chaîne de caractères y à tester avec la fonction **Palindrome** devra être : *engagelejeuquejelegagne* ;
- teste si y est un palindrome à l'aide de la fonction **Palindrome** ;
- affichage du résultat du test.

PROBLEME (coefficient 5)

On veut réaliser des opérations de tri sur le contenu d'un tableau T qui comporte 2 colonnes et autant de lignes que de produits d'un magasin. La première colonne correspond aux numéros de référence des produits. La deuxième colonne correspond à la quantité de chaque produit en stock. Par exemple, si le magasin possède 5 produits, le tableau T aura la structure suivante :

9109	56
1890	39
3120	21
5640	13
8698	10

Il vous est demandé de concevoir les trois procédures suivantes :

1. Une procédure **Remplis** de remplissage du tableau T pour N produits. Les paramètres de la procédure seront le nombre de lignes N et le tableau T . Définissez pour cela dans le programme principal un nouveau type TAB qui correspondra à un tableau à deux dimensions (100 lignes et 2 colonnes) :

TYPE TAB = ...

2. Une procédure **Tri** qui effectue le tri du tableau par numéro de référence croissant (tri sur la première colonne, variable $num = 1$) :

1890	39
3120	21
5640	13
8698	10
9109	56

ou par quantité en stock croissante (tri sur la deuxième colonne, variable $num = 2$) :

8698	10
5640	13
3120	21
1890	39
9109	56

Le numéro de colonne num sur laquelle s'opère le tri doit être traité comme un paramètre de la procédure **Tri**. L'algorithme utilisé pour effectuer le tri peut être au choix celui du tri par insertion ou celui du tri par sélection.

3. Une procédure **Quantite** qui affiche la quantité en stock d'un produit de référence donnée ref . Cette procédure aura pour paramètres : le nombre de produits N , le tableau T et la référence recherchée ref .

Pour terminer, écrire le programme principal qui, après avoir fait appel à la procédure **Remplis**, propose au choix : de trier le tableau par numéro de référence croissant ou par quantité en stock croissante à l'aide de la procédure **Tri** ou bien encore d'afficher la quantité en stock d'un produit de référence ref en faisant appel à la procédure **Quantite**.